New Constraints on Generation of Uniform Random Samples from Evolutionary Trees

Conference Paper *in* Canadian Conference on Electrical and Computer Engineering · May 2006

DOI: 10.1109/CCECE.2006.277581 · Source: DBLP

Kazi Zakia Sultana

Montclair State University

28 PUBLICATIONS **107** CITATIONS

NEW CONSTRAINTS ON GENERATION OF UNIFORM RANDOM SAMPLES FROM EVOLUTIONARY TREES

Anupam Bhattacharjee
Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh.
email: abrbuet@yahoo.com

Zalia Shams
Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh.
email: setu_18@yahoo.com

Kazi Zakia Sultana
Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh.
email: z_sultana00@yahoo.com

Abstract

In this paper, we introduce new algorithms for selecting taxon samples from large evolutionary trees, maintaining uniformity and randomness, under certain new constraints on the taxa. The algorithms are efficient as their runtimes and space complexities are polynomial. The algorithms have direct applications to the evolution of phylogenetic tree and efficient supertree construction using biologically curated data. We also present new lower bounds for the problem of constructing evolutionary tree from experiment under some earlier stated constraints. All the algorithms have been implemented.

***Keywords**: Uniformity, randomness, evolutionary tree, polynomial algorithms, NP-hard problems, bioinformatics.*

1. Introduction

One of the applications of trees on the field of bioinformatics is *evolutionary tree*. A well-studied problem in computational biology is the accurate reconstruction of evolutionary trees from sequence or distance data. An evolutionary tree is a tree where the leaves represent *species* and internal nodes represent common ancestors.

There are many different approaches to the problem of constructing an evolutionary tree. From the distance-based methods such as *Fitch-Margoliash* and *Neighbor-Joining*, to more recent *Quartet*, and *Expectation-Maximization* methods, there is a wide variety of techniques used to generate phylogenetic topologies. The problem is known to be NP-hard.

A well-known variant of the problem of constructing an evolutionary tree is based on experiments that determine how three species are related in the evolutionary tree. The approach involves generating a topology on the taxa using model, then generating a sequence label for the root and evolving the sequence along the paths in the tree to obtain a label for each leaf [1]. Efficiency and performance of the algorithm depends on how close the inferred tree and the target tree are.

The problem of constructing evolutionary trees using experiments has been studied by Kannan, Lawler, and Warnow [2]. They show that as the number of experiment decreases the running times of the algorithms increase. Kao, Lingas and Ostlin propose an efficient algorithm for the problem and its variations [2]. Their algorithm works very efficiently using *balanced randomized tree splitting* technique. Using their algorithm an evolutionary tree for species can be determined, using experiment, in expected time $O(nd \log n \log \log n)$, d denotes the maximum degree in the tree [3].

Related to the sampling problem is another important problem in Bioinformatics that concerns the construction of *supertrees* from a set of phylogenetic trees [4]. There are also a wide variety of methods for supertree construction including matrix representation of parsimony (*MRP*), *BUILD*, and *quartet* methods. Kearney, Munro, and Phillips [1] introduce three nontrivial constraints for the sampling process and solve them. We, in this paper, by using different better approaches, improve the runtime of their algorithms. We also present three new constraints in this paper and solve them.

2. The Old Constraints

Kearney, Munro, and Phillips introduce algorithms for sampling subsets of taxa from a large phylogenetic tree, subject to three biologically motivated constraints. The first constraint, called *leaf-depth constraint*, specifies a range of depth for a leaf in the induced subtree. This transfers to a constraint on the amount of evolution that each taxon has experienced since the existence of this set of taxa. The second constraint, the *edge-length constraint*, specifies a range of edge length for every edge in the induced subtree. Significant disparity in edge length can cause difficulties for some phylogeny reconstruction algorithm. The last constraint that they introduce is the *pairwise leaf distance constraints*. It is useful as very close and very distant sequences can be problematic.

To solve the problems, they use *balanced search trees*. We, in this paper, introduce the approach of *memorization* and *hashing* to reduce the total running time of the algorithm with a very little change in space complexity. Moreover, we introduce three new constraints that are similar to the old three but more nontrivial. The first of them, hereafter called the *pairwise leaf distance exclusion constraint*, specifies a construction scheme of samples where distance between any two leaves excludes a certain range. It is useful to detect *slow genome evolution* and *rapid genome evolution* and to cluster species in different close groups. The second constraint is the *pairwise leaf distance multiple range exclusion* (shortly, the *multiple range exclusion*) *constraint* according to which the pairwise distances should exclude the given ranges. The constraint has similar significance as the first one. The last constraint, called *certain pair distance constraint*, specifies a construction approach where distance between a certain pair of species maintains a range. The constraint is useful when we want to sample based on a known range between a certain pair of species. The algorithms we developed can also be used to test the performance of supertree algorithms systematically.

Problem. *(Leaf Depth): Given a phylogenetic tree T with n leaves, a positive integer p, a positive integer $m \geq 2$, and two non-negative real numbers $d_{\min}$ and $d_{\max}$, select p samples of m leaves from T, uniformly at random from all sets of size m such that if r is the root of the subtree induced by a sample then for every leaf v in the sample $d_{\min} < dist(r,v) < d_{\max}$.*

To solve this problem, our algorithm begins enumerating every sample of size m in the tree that satisfies the given constraint and then uses this information to generate uniform samples. Enumeration can be carried out by storing valid leaf information at each node u in the tree. Now, we can examine every node u in T as the potential root of a sample and determine how many valid samples are rooted here .If there are V_L valid leaves in left subtree and V_R in the right, then S_u, the number of valid samples rooted at u is given by:

$$S_u = \binom{V_L+V_R}{m} - \binom{V_L}{m} - \binom{V_R}{m}$$

The probability of a valid sample being rooted at a node u is $S_u \div S$, where S is the total number of valid samples in T. Using these probabilities, we choose a root r and m valid leaf descendants of r, uniformly at random, ensuring at least one descendant from both the left and the right child of r. Using the algorithm, we can establish the following result:

Theorem 1. *The Leaf Depth problem can be solved in $\theta(n)$ time and $\theta(n)$ space.*

Lemma 1. *Given a phylogenetic tree T with n leaves, the number of valid leaf descendants can be enumerated in $\theta(n)$ time and space.*

Proof: Using memoization, we can traverse the whole tree in $\theta(n)$ time. Just returning from the leaves, we can calculate the values of V_L and V_R. These calculated values can be used from all the ancestors on the same purpose. No further re-computation is needed. So, enumeration of the number of valid samples for each node in T takes $\theta(n)$ time.

We need to store is only the tree and S_U vector per node. So, we need a data structure of space only $\theta(3n) \approx \theta(n)$.

Lemma 2. *Given S_U, the no. of valid samples of size m rooted at u, for all nodes u in T, a uniform random valid sample rooted at u can be chosen in $\theta(n)$ time using $\theta(n)$ space.*

Proof: The previous technique is first to choose a root in $\theta(n)$ time by assigning $S_u \div S$ for each node u. Instead, we can do here a binary search to choose a root r, closest to the given probability in the range $[0,...,1]$ and the process needs $\theta(\log_2 n)$ time. After choosing the root we traverse both the left and right subtree adding a valid leaf in our sampled tree. So the total work can be performed in $\theta(n)$ time and $\theta(n)$ space(for choosing root r). Thus, *Theorem 1* comes from *Lemma 1* and *Lemma 2.*

Problem. *(Edge Length) Given a phylogenetic tree T with n leaves, a positive integer p, a positive integer $m \geq 2$, and two non-negative real numbers $e_{\min}$ and $e_{\max}$ select p samples of m leaves from T, uniformly at random from all sets of size m such that for every edge e in the subtree induced by the sample, $e_{\min} < |e| < e_{\max}$.*

The basic idea is to consider each node u in the tree as a potential root and then find ways for combining valid samples of size $j = 1,...,m-1$ rooted at some node in the left subtree of u with valid samples of size $m-j$ rooted at a node in the right subtree, ensuring the constraint related to every edge.

Theorem 2. *The Edge Length problem can be solved in $\theta(m^2 n)$ time and $\theta(mn)$ space.*

Lemma 3. Given a phylogenetic tree T with n leaves, a positive integer $m \geq 2$, and two non-negative real numbers $e_{\min}$ and $e_{\max}$, the number of valid samples of size $j = 1,...,m$ rooted each node in T can be enumerated in $\theta(m^2 n)$ time using $\theta(mn)$ space.

Proof. $S_U[k]$ is number of valid samples of size $k = 1,...,m$ rooted at u and $S_U[k] = \sum_{i=1}^{k-1} S^L[i] \times S^R[k-i]$.

Here, $S^L[k]$ and $S^R[k]$ for $k = 1,...,m-1$ represent the number of valid samples of size k rooted at a valid left and right descendant of u, respectively. The total number of descendants in tree on n leaves is at most nh (h=height of T). These values can be obtained for each node in total time $\theta(mnh)$. But by using memoization and not storing the ordered descendancy list, we reduce the cost to $\theta(mn)$. $S_U[k]$ is calculated in $\theta(m)$ time. Finding the values $S_U[k]$, for all k and u, will cost $\theta(m^2n)$. Now, total enumeration time is $\theta(mn+m^2n)\approx\theta(m^2n)$. At every node, we store a vector of size m. So, the total space required is $\theta(mn)$.

Lemma 4: Given a node r and an integer $m \geq 1$, a uniform random valid sample of size m rooted at r can be chosen in $\theta(mn)$ time and $\theta(mn)$ space.

Proof. By induction on m, if $m=1$, the only valid samples are rooted at a leaf. Thus, r must be a leaf and so our sample consists simply of r. If $m>1$, we traverse the left and right subtrees rooted at r to find the number of valid samples of size $1,\ldots,m-1$ rooted at all descendants u of r for which $e_{\min} < dist(r,u) < e_{\max}$ and store these values in the vectors V_L and V_R, respectively.

We will take i ($1 \leq i \leq m-1$) leaves from the left subtree , while the rest are taken from the right subtree. There are $V_L[i]\times V_R[m-i]$ valid samples of size m with i taken from the left subtree. Now, we find correct descendants in the left subtree of r and recursively pick a valid sample rooted there of size i. Similarly we obtain a sample of size $m-i$ from the right subtree.

For each internal node, in $\theta(m)$ time we decide how many leaves to take from the left subtree and then $\theta(n)$ time to find the correct descendant nodes. There will be $m-1$ internal nodes in the induced subtree, so the total time required is $\theta((m-1)(m+n))\approx\theta(mn)$ which is an improvement over the earlier paper.The required space stores the V_L and V_R vectors, which have at most $m-1$ entries each. *Theorem 2* follows from *Lemma 3* and *Lemma 4*.

Problem. *(Pairwise Leaf Distance) Given a phylogenetic tree T with n leaves, a positive integer p, a positive integer $m \geq 2$, and two non-negative real numbers $d_{\min}$ and $d_{\max}$, select p samples of m leaves from T, uniformly at random from all samples of size m such that the pairwise distance between any two leaves in the sample is greater than $d_{\min}$ and smaller than $d_{\max}$.*

Theorem 3. The Pairwise Leaf Distance problem can be solved in $O(m^2n^5+pm^2n^3)$ time and $O(m^2n^3)$ space.

Proof: The proof is omitted for space shortage.

3. The New Constraints

Problem. *(Pairwise Leaf Distance Exclusion) Given a phylogenetic tree T with n leaves, a positive integer p, a positive integer $m \geq 2$, and two non-negative real numbers $d_{\min}$ and $d_{\max}$, select p samples of m leaves from T, uniformly at random from all samples of size m such that the pairwise distance between any two leaves in the sample is less than $d_{\min}$ and greater than $d_{\max}$.*

During the enumeration, We calculate S_U, the no of valid samples of size m rooted at u, which is used to select the root r of a sample.

Theorem 4: The pairwise leaf distance exclusion constraint can be solved in $O(mn^2+n^2)$ time and $O(n^2)$ space.

Lemma 5. Let, u be a node in T. Total number of trees rooted at u satisfying the given constraint can be obtained in time $O(mn^2)$ using $O(n^2)$ space.

Proof: A root maintains a *doubtful list* and a *safe counter*. If the distance between this root and a leaf is less than the maximum value of the given range, the distance will be stored in doubtful list. If addition of edge lengths make the pairwise distance fall above the given range we exclude this node from doubtful list and add to safe counter. At the end those items having pairwise distance less than lower limit are excluded from doubtful list and added to safe counter. Using memoization, $O(mn^2)$ time will be needed to compute valid number of trees for all nodes. Again with a list of size at most n at each node, the space needed is $O(n^2)$.

Lemma 6: Given a root r for the subtree induced by a sample, a valid sample rooted at r can be generated uniformly at random in $O(n^2)$ time.

Proof: Choosing a root we traverse it's children in the same way to construct the whole tree. For a node, the doubtful list of both the children is searched to exclude invalid combinations. This requires $O(n)$ operations. So, the total time is $O(n^2)$. The proof of *Theorem 4* follows from *Lemma 5* and *Lemma 6.*

Problem. *(Multiple Range Exclusion) Given a phylogenetic tree T with n leaves, a positive integer p, a positive integer $m \geq 2$, and a list of ranges, select p samples of m leaves from T, uniformly at random from all samples of size m, such that the pairwise distance between any two leaves in the sample is beyond any of the given ranges.*

We have to maintain two lists: *safe list* and *doubtful list*. The cardinality of safe list denotes the enumeration. l denotes the number of ranges given to be excluded during sampling.

Theorem 5: The multiple range constraint can be solved in time in $O(mn^2l+n^2l)$ time and $O(n^2)$ space.

Lemma 7. Let, u be a node in T. Total number of valid trees rooted at u satisfying the given constraint can be obtained in time $O(mn^2l)$ using space $O(n^2)$.

Proof: The same algorithm as the pairwise leaf distance exclusion constraint is used here but an extra loop is needed to check all the given ranges to maintain doubtful list and safe list. Using memoization, $O(mnl)$ time is needed to compute valid number of trees at each node. So, cost will be increased to $O(mn^2l)$. The space needed to keep the list is $O(n^2)$.

Lemma 8: Given a root r for the subtree induced by a sample, a valid sample rooted at node r can be generated uniformly at random in $O(n^2l)$ time.

*Proof: Like pairwise leaf d*istance exclusion constraint, each node takes $O(nl)$ time. So the total time required is $O(n^2l)$.

The proof of *Theorem 5* follows from *Lemma 7* and *Lemma 8*.

Problem. (*Certain Pair Distance*) Given a phylogenetic tree T with n leaves, a positive integer $m \geq 2$ and two non-negative real numbers $d_{\min}$ and $d_{\max}$, select p samples of m leaves from T uniformly at random from all samples of size m such that for a certain pair of leaves, let (x, y) and if both x and y are present in the sample, $d_{\min} < dist(x, y) < d_{\max}$.

In the supertree two cases may arise: both x and y exist or not. For the former case we have to check the distance. For the later case, all samples of m leaves are valid.

Let us assume that, if both x and y exist x is the leaf in the left subtree and y in the right subtree of the root.

We keep five measures at a node u in T:

1. Distance from x (infinite if x does not exist in the tree considering u as the root),
2. Distance from y similarly,
3. Number of valid samples rooted at u excluding x, S_{ux}.
4. Number of valid samples rooted at u excluding y, S_{uy}.
5. Total number of valid samples S_u.

In a node where S_u or S_{uy} has no particular meaning, we equal them to S_u. S_u^L and S_u^R are the calculated S_u in the left and right children of u respectively. Similar meanings are used for S_{ux}^L and S_{uy}^R.

Now approaching with memoization to calculate the five values, following cases may arise at a node u:

1. Neither the left subtree contains x nor the right subtree contains y, or both x and y belong to the same subtree. In such cases, $S_u = S_{ux} = S_{uy} = S_u^L \times S_u^R$.
2. a) x exists but y dosen't:
   $$S_u = S_{uy} = S_u^L \times S_u^R,\ S_{ux} = S_{ux}^L \times S_u^R.$$
   b) y exists but x doesn't:
   $$S_u = S_{ux} = S_u^L \times S_u^R,\ S_{uy} = S_u^L \times S_{uy}^R.$$
3. x belongs to left subtree and y belongs to the right subtree rooted at u.

Let, d = distance from x + distance from y. Now, if $d_{\min} < d < d_{\max}$ then, x and y maintain the constraint, $S_u = S_{ux} = S_{uy} = S_u^L \times S_u^R$.

Otherwise, $S_u = S_{ux} = S_{uy} = S_{ux}^L \times S_u^R + S_u^L \times S_{uy}^R$.

Theorem 6. *The Certain Pair Distance problem can be solved in $\theta(n)$ time and $\theta(n)$ space.*

Lemma 9. *Given a phylogenetic tree T with n leaves, the number of valid leaf descendants can be enumerated in $\theta(n)$ time and space.*

Proof. We apply the same technique used in Lemma 1 to prove this Lemma. The space required is $\theta(n)$ as what we need to store is only the tree and S_U vector per node. So, we need a data structure of space only $\theta(7n) \approx \theta(n)$.

Lemma 10. *Given S_U, the no of valid samples of size m rooted at u for all nodes u in T, a uniform random valid sample can be chosen in $\theta(n)$ time using $\theta(n)$ space.*

Proof. Omitted for space shortage.

The proof of *Theorem 6* follows from *Lemma 9* and *Lemma 10.*

4. Conclusion

In this paper, we present some improved algorithms to solve the leaf depth constraint, the edge length constraint, and the pairwise leaf distance constraint. We, in addition, introduce three new constraints, the pairwise leaf distance exclusion constraint and multiple range exclusion constraint, and present algorithms with tight lower bounds of runtimes and space complexities to solve them.

References

[1] P. Kearney, J. I. Munro, and D. Phillips, "Efficient generation of uniform samples from phylogenetic trees", *Wabi 2003*, LNBI 2812, Springer-Verlag, 177-189, 2003.

[2] S. K. Kannan, E. L. Lawler, and T. J. Warnow, "Determining the evolutionary tree using experiments", *Journal of Algorithms*, 21:26-50, 1996.

[3] D. Phillips, "Uniform Sampling from phylogenetic trees", *Masters Thesis*, University of Waterloo, August 2002.

[4] N. C. Jones, and P. A. Pevzner, "*AN INTRODUCTION TO BIOINFORMATICS ALGORITHMS*", *The MIT Press*, Cambridge, Massachusetts, London, England, 2004.